Appendix B: Check For Side Effects

```python
import re
import sys
sys.path.append("./utils")

# import local constants and helpers. See Appendix C for Keywords.
from keywords import side_effect_keywords, negation_words
# See Appendix D for parser.
from parser import parse_string, find_first_occurence

def check_for_negation(text, keyword):
    words = text.split()
    keyword_index = find_first_occurence(keyword, words)

    if keyword_index == -1:
        return False

    # Search for negation within the 3 words before the keyword
    window = words[max(0, keyword_index - 3): keyword_index]

    # iterate through the negation words and check if any of them are found in the
window.
    for negation in negation_words:
        if any(negation in word for word in window):
            return True
    return False

# Create a function to check if the review mentions side effects or no side effects
def check_side_effects(review):
    if review == '':
        return 0

    review = parse_string(review) # Convert the review to lowercase

    # Search for any of the keywords in the review
    for keyword in side_effect_keywords:
        # Search for the keyword as a whole word using regular expressions
        if re.search(re.escape(keyword), review):
            # If the keyword is found, check for negation
            if not check_for_negation(review, keyword):
                return 1  # Side effect found and not negated
    return 0  # No side effects mentioned or negated
```

Appendix C: Keywords

```python
side_effect_keywords = [
    # General symptoms
    'headache', 'migraine', 'dizzy', 'dizziness', 'lightheaded', 'faint', 'nausea',
'nauseous',
    'vomit', 'vomiting', 'upset stomach', 'stomach ache', 'abdominal pain', 'cramps',
'bloating',
    'diarrhea', 'constipation', 'indigestion', 'heartburn', 'gas', 'belching', 'burping',

    # Fatigue and energy
    'fatigue', 'tired', 'exhausted', 'sleepy', 'drowsy', 'lethargic', 'low energy',
'sluggish',
    'weakness', 'muscle weakness', 'muscle pain', 'joint pain', 'achy', 'aches', 'pain',
'back pain',

    # Mood and mental health
    'anxiety', 'anxious', 'panic attacks', 'depression', 'mood swings', 'irritable',
'agitation',
    'nervous', 'restless', 'confusion', 'brain fog', 'trouble concentrating',
'forgetfulness', 'sadness',

    # Sleep-related issues
    'insomnia', 'trouble sleeping', 'difficulty sleeping', 'cant sleep', 'restless sleep',
'waking up at night',
    'nightmares', 'vivid dreams', 'sleep disturbance', 'oversleeping', 'sleep too much',
'sleep paralysis',

    # Skin-related issues
    'rash', 'hives', 'itch', 'itching', 'redness', 'dry skin', 'flaky skin', 'acne',
'breakouts',
    'eczema', 'swelling', 'inflammation', 'puffy', 'puffiness', 'skin irritation',
'bruising',

    # Weight and appetite
    'weight gain', 'weight loss', 'loss of appetite', 'increased appetite', 'overeating',
'cravings',
    'binge eating', 'bloating', 'water retention', 'swelling in legs', 'fluid retention',

    # Cardiovascular-related
    'palpitations', 'heart racing', 'fast heartbeat', 'slow heartbeat', 'chest pain', 'chest
tightness',
    'high blood pressure', 'low blood pressure', 'hypertension', 'fainting', 'dizziness
standing up',

    # Vision and eyes
    'blurry vision', 'vision changes', 'eye pain', 'dry eyes', 'watery eyes', 'sensitive to
light', 'double vision',

    # Mouth, throat, and breathing
    'dry mouth', 'cotton mouth', 'thirsty', 'sore throat', 'cough', 'difficulty breathing',
'shortness of breath',
```

```python
    'wheezing', 'congestion', 'runny nose', 'stuffed nose', 'nosebleeds', 'difficulty
swallowing', 'hoarse voice',

    # Sexual and reproductive system
    'loss of libido', 'low sex drive', 'erectile dysfunction', 'impotence', 'painful
intercourse', 'vaginal dryness',
    'menstrual changes', 'irregular periods', 'missed period', 'spotting', 'cramps', 'heavy
periods', 'prolonged periods',
    'discharge', 'brown discharge', 'breast tenderness', 'breast pain', 'nipple soreness',

    # Allergy-related
    'allergic reaction', 'swelling of the face', 'swelling of lips', 'swelling of tongue',
'difficulty breathing',
    'hives', 'itching', 'anaphylaxis', 'rash', 'throat swelling',

    # Neurological and cognitive
    'tremors', 'shaking', 'numbness', 'tingling', 'pins and needles', 'seizures',
'convulsions', 'loss of balance',
    'dizziness', 'vertigo', 'coordination issues', 'memory loss', 'confusion',

    # Urinary and renal system
    'frequent urination', 'urgent urination', 'difficulty urinating', 'painful urination',
'kidney pain', 'dark urine',
    'blood in urine', 'urinary tract infection', 'UTI', 'incontinence', 'trouble holding
urine',

    # Miscellaneous
    'fever', 'chills', 'sweating', 'night sweats', 'hair loss', 'brittle nails', 'ringing in
the ears', 'tinnitus',
    'bad taste', 'metallic taste', 'dry lips', 'swollen lips', 'joint stiffness'
]

# Common negation words or phrases
negation_words = [
    # Single-word negations
    "no", "not", "never", "none", "nothing", "nowhere", "neither", "nobody", "without",
"hardly", "barely", "scarcely",

    # Common contractions
    "dont", "doesnt", "didnt", "wont", "wouldnt", "isnt", "arent", "wasnt", "werent",
"havent", "hasnt",
    "hadnt", "cant", "couldnt", "shouldnt", "mustnt", "mightnt", "shant", "aint",

    # Colloquial and informal negations
    "aint", "nah", "nope", "nothin", "not a single", "not any", "didnt", "doesnt", "havent",
"isnt", "arent", "wasnt", "werent",

    # Phrases indicating negation
    "by no means", "in no way", "on no account", "under no circumstances", "neither nor", "no
longer", "no way", "not at all",
    "not even", "not once", "not anymore", "far from", "seldom", "few if any", "least of all"
]
```

Appendix D: Parser

```python
import re
import urllib.parse
import html

def parse_string(input):
    # Decode HTML entities and URL-encoded characters in a single pass
    input = html.unescape(urllib.parse.unquote(input))

    # remove excess white space and convert hyphens to spaces
    input = re.sub(r'[\-]+|\s+', ' ', input).strip()

    # remove any special characters
    input = re.sub(r'[^a-zA-Z0-9 ]+', '', input)

    return input.lower()

def find_first_occurence(key, arr):
    if key == '':
        return -1

    for (i, item) in enumerate(arr):
        # if the item is 3 characters or less, skip.
        # else if there is a match found
        if len(item) <= 3:
            continue
        elif item in key or key in item:
            return i

    return -1
```